|  |
| --- |
| **Function design** |
| **libharkio3 Library** |
|  |
| Libharkio3の関数仕様書 |
|  |
| **takenaka** |
| **2013/06/14** |
|  |

目　次

[**構造体一覧表** 6](#_Toc380657206)

[**1.** **enum** 6](#_Toc380657207)

[**2.** **multipleType** 6](#_Toc380657208)

[**3.** **harkio\_Matrix** 6](#_Toc380657209)

[**4.** **harkio\_Position** 7](#_Toc380657210)

[**5.** **harkio\_Config** 8](#_Toc380657211)

[**6.** **harkio\_Positions** 8](#_Toc380657212)

[**7.** **harkio\_Neighbors** 8](#_Toc380657213)

[**8.** **harkio\_XML** 9](#_Toc380657214)

[**9.** **harkio\_TransferFunction** 9](#_Toc380657215)

[**Matrix** 10](#_Toc380657216)

[1. harkio\_Matrix\_new　（新規harkio\_Matrix」型構造体作成） 11](#_Toc380657217)

[2. harkio\_Matrix\_ones　（すべての要素が 1 の配列を作成） 12](#_Toc380657218)

[3. harkio\_Matrix\_ zeros　（すべての要素が 0 の配列を作成） 13](#_Toc380657219)

[4. harkio\_Matrix\_eye　（対角要素が 1 でその他の要素が 0 の m 行 n 列の行列を作成） 14](#_Toc380657220)

[5. harkio\_Matrix\_delete　（作成したhark\_Matrix型構造体のメモリを削除（解放）） 15](#_Toc380657221)

[6. harkio\_Matrix\_setValueInt32　（int型の行列の指定箇所（row , col）にvalue値をセットする） 16](#_Toc380657222)

[7. harkio\_Matrix\_getValueInt32　（int型の行列の指定箇所から要素を取り出しvalueに設定） 17](#_Toc380657223)

[8. harkio\_Matrix\_setValueFloat32　（float型の行列の指定箇所（row , col）にvalue値をセットする） 18](#_Toc380657224)

[9. harkio\_Matrix\_getValueFloat32　（float型の行列の指定箇所から要素を取り出しvalueにセット） 19](#_Toc380657225)

[10. harkio\_Matrix\_setValueComplex（float型の行列にvalue値をセットする） 20](#_Toc380657226)

[11. harkio\_Matrix\_getValueComplex32　（complex型の行列から要素を取り出しvalueにセット） 21](#_Toc380657227)

[12. harkio\_Matrix\_getValues　（指定範囲の要素を取得する） 22](#_Toc380657228)

[13. harkio\_Matrix\_setValues　（Matrix型(複数要素)をセットする） 23](#_Toc380657229)

[14. harkio\_Matrix\_add　（行列aと行列bの各要素の和） 24](#_Toc380657230)

[15. harkio\_Matrix\_sub　（行列aと行列bの各要素の差） 25](#_Toc380657231)

[16. harkio\_Matrix\_mul　（行列の積） 26](#_Toc380657232)

[17. harkio\_Matrix\_emul　（行列の各要素ごとの積） 27](#_Toc380657233)

[18. harkio\_Matrix\_sum　（行列の合計） 28](#_Toc380657234)

[19. harkio\_Matrix\_fromFile　（ファイルの内容をharkio\_Matrix型構造体にセットする） 29](#_Toc380657235)

[20. harkio\_Matrix\_print　（harkio\_Matrix構造体の内容をすべて表示する） 30](#_Toc380657236)

[21. harkio\_Matrix\_write　（harkio\_Matrix型構造体の内容をファイルに書き出す） 31](#_Toc380657237)

[22. harkio\_Matrix\_isEqual　（行列比較） 32](#_Toc380657238)

[23. harkio\_Matrix\_size　（行数と列数を、size配列にセットする） 33](#_Toc380657239)

[24. harkio\_Matrix\_ fromZip（zipファイルからharkio\_Matrixに変換する） 34](#_Toc380657240)

[**XML** 35](#_Toc380657241)

[25. harkio\_XML\_new　（新規harkio\_XML型構造体作成） 36](#_Toc380657242)

[26. harkio\_XML\_newFromXML（XMLファイルの内容を「harkio\_XML」型構造体に設定） 37](#_Toc380657243)

[27. harkio\_XML\_getTag　(tag(version)を取得する) 39](#_Toc380657244)

[28. harkio\_XML\_getType　(typeを取得する) 40](#_Toc380657245)

[29. harkio\_XML\_getRow　(rowを取得する) 42](#_Toc380657246)

[30. harkio\_XML\_getCol　(colを取得する) 43](#_Toc380657247)

[31. harkio\_XML\_newFromData　（入力パラメータから、新規に「harkio\_XML」型構造体を作成） 44](#_Toc380657248)

[32. harkio\_XML\_delete　(Harkio\_XML型構造体の領域を開放する) 45](#_Toc380657249)

[33. harkio\_XML\_getNeighbors　（近接ID情報を取得する） 46](#_Toc380657250)

[34. harkio\_XML\_getPositions　(位置情報を取得する) 47](#_Toc380657251)

[35. harkio\_XML\_setPositions　(位置情報をセットする) 48](#_Toc380657252)

[36. harkio\_XML\_getConfig　(位置情報を取得する) 49](#_Toc380657253)

[37. harkio\_XML\_write　(XML形式で出力する) 50](#_Toc380657254)

[38. harkio\_XML\_writeBuffer　(XML形式に変換してバッファに出力する) 51](#_Toc380657255)

[39. harkio\_XML\_internal\_write　(XML形式に出力する。) 52](#_Toc380657256)

[40. harkio\_XML\_internal\_processNode　（読み込んだデータを、「harkio\_XML」型構造体に変換する） 53](#_Toc380657257)

[41. harkio\_XML\_internal\_strcount　(文字列の総数を取得する) 54](#_Toc380657258)

[42. pushStream　（Streamに出力する） 55](#_Toc380657259)

[**Position (位置)** 56](#_Toc380657260)

[**43.** **harkio\_Position\* harkio\_Position\_new（新規に「harkio\_Position」型構造体を作成する）** 57](#_Toc380657261)

[**44.** **harkio\_Position\_delet (「harkio\_Position」型構造体を削除する)** 58](#_Toc380657262)

[**45.** **float\* harkio\_Position\_getPosition（座標を取得する）** 59](#_Toc380657263)

[**46.** **harkio\_Position\_setPosition（Positionをセットする）** 60](#_Toc380657264)

[**47.** **harkio\_Position\_setChannels（channelsをセットする）** 61](#_Toc380657265)

[**48.** **harkio\_Position\_setChannels（channelsをセットする）** 62](#_Toc380657266)

[**49.** **harkio\_Position\_setNumChannels（numChannelsをセットする）** 63](#_Toc380657267)

[**50.** **harkio\_Position\_setChannelsUse（ChannelsUseをセットする）** 64](#_Toc380657268)

[**51.** **harkio\_Position\_getID (idを取得する)** 65](#_Toc380657269)

[**52.** **harkio\_Position\_getSys (sys(coordinate)を取得する)** 66](#_Toc380657270)

[**53.** **harkio\_Position\_getPath (Pathを取得する)** 67](#_Toc380657271)

[**54.** **harkio\_Position\_print（Positionを表示する）** 68](#_Toc380657272)

[**55.** **harkio\_Position\_isEqual（ hark\_positionの比較）** 69](#_Toc380657273)

[**56.** **harkio\_Position\_copy（オブジェクトのコピーを行う）** 70](#_Toc380657274)

[**57.** **harkio\_Position\_Convert（座標変換する）** 71](#_Toc380657275)

[**58.** **harkio\_Position\_convToRadian (角度(degree)からラジアン(radian)に変換する)** 72](#_Toc380657276)

[**59.** **harkio\_Position\_convToRadian (ラジアン(radian)から角度(degree)に変換する)** 73](#_Toc380657277)

[**Positions (位置の集合)** 74](#_Toc380657278)

[60. harkio\_Positions\_new（新規harkio\_Positions型構造体作成） 75](#_Toc380657279)

[61. harkio\_Positions\_delete(harkio\_Positions型領域を削除) 76](#_Toc380657280)

[**62.** **harkio\_Positions\_getById (入力したIDと同じIDのharkio\_Positionを返す)** 77](#_Toc380657281)

[**63.** **harkio\_Positions\_getByCoord (入力パラメータからHark\_positionを取得する)** 78](#_Toc380657282)

[**64.** **harkio\_Positions\_getType (Typeを取得する)** 79](#_Toc380657283)

[**65.** **harkio\_Positions\_getSize (sizeを取得する)** 80](#_Toc380657284)

[**66.** **harkio\_Positions\_getFrame (frameを取得する)** 81](#_Toc380657285)

[**67.** **harkio\_Positions\_readFiles (ファイルから読み込む)** 82](#_Toc380657286)

[**68.** **harkio\_Positions\_append（Hark\_positonsにhark\_positionを追加する）** 83](#_Toc380657287)

[**69.** **harkio\_Positions\_removeById (削除対象id のpositionを削除する)** 84](#_Toc380657288)

[**70.** **harkio\_Positions\_removeByCoord (削除対象座標のpositionを削除する)** 85](#_Toc380657289)

[**71.** **Positions\* harkio\_Positions\_copy（引数のhark\_Positinsのコピーを作成する）** 87](#_Toc380657290)

[**72.** **harkio\_Positions\_isEqual (Positionを比較する)** 88](#_Toc380657291)

[**73.** **harkio\_Positions\_print (hark\_positionsの表示をする)** 89](#_Toc380657292)

[**Neighbors (隣接位置)** 90](#_Toc380657293)

[**74.** **harkio\_Neighbors\_new (新規で、harkio\_Neighbors型を作成)** 91](#_Toc380657294)

[**75.** **harkio\_Neighbors\_delete (harkio\_Neighbors型領域を削除)** 92](#_Toc380657295)

[**76.** **harkio\_Neighbors\_calcNeighbors (アルゴリズムで取得した結果を入力パラメータに設定)** 93](#_Toc380657296)

[**77.** **harkio\_Neighbors\_append　（Hark\_positonsにhark\_positionを追加する）** 94](#_Toc380657297)

[**78.** **harkio\_Neighbors\_Algorithms\_NearestNeighbor (NearestNeighboを用いて設定する)** 96](#_Toc380657298)

[**79.** **harkio\_Neighbors \_getSize(sizeを取得する)** 97](#_Toc380657299)

[**80.** **harkio\_Neighbors \_getNeighborAlgorithm(algを取得する)** 98](#_Toc380657300)

[**81.** **euclidDistance** 99](#_Toc380657301)

[**82.** **harkio\_Neighbors\_print (harkio\_Neighborsを表示する)** 100](#_Toc380657302)

[**83.** **harkio\_Neighbors\_isEqual (harkio\_Neighborsを比較する)** 101](#_Toc380657303)

[**84.** **harkio\_Neighbors\_copy　(harkio\_Neighborsをコピー)** 102](#_Toc380657304)

[**TransferFunction (伝達関数操作)** 103](#_Toc380657305)

[85. harkio\_TransferFunction\_new (harkio\_TransferFunction型領域を新規作成する) 104](#_Toc380657306)

[**86.** **harkio\_TransferFunction\_fromData　（）** 105](#_Toc380657307)

[**87.** **harkio\_Neighbors\_delete　（harkio\_Neighbors型領域を削除）** 106](#_Toc380657308)

[**88.** **harkio\_TransferFunction\_fromFile　（ZIPファイルから、伝達関数を読み込む）** 107](#_Toc380657309)

[**89.** **harkio\_TransferFunction\_getConfig (harkio\_TransferFunctionからHark\_Configを取得する)** 108](#_Toc380657310)

[**90.** **harkio\_TransferFunction\_getNeighbors (harkio\_Neighborsを取り出す)** 109](#_Toc380657311)

[**91.** **harkio\_TransferFunction\_getMicrophones (harkio\_TransferFunctionからmic座標を取り出す)** 110](#_Toc380657312)

[**92.** **harkio\_TransferFunction\_getPositions (harkio\_TransferFunctionから位置情報を取り出す)** 111](#_Toc380657313)

[**93.** **harkio\_TransferFunction\_setConfig (harkio\_TransferFunctionにharkio\_Configをセットする)** 112](#_Toc380657314)

[**94.** **harkio\_TransferFunction\_setNeighbors (harkio\_Neighborsをセットする)** 113](#_Toc380657315)

[**95.** **harkio\_TransferFunction\_setMicrophones (harkio\_TransferFunctionにマイク位置をセットする)** 114](#_Toc380657316)

[**96.** **harkio\_TransferFunction\_setPositions (harkio\_TransferFunctionに、harkio\_Positionsを設定する)** 115](#_Toc380657317)

[**97.** **harkio\_TransferFunction\_setLocTFs (harkio\_TransferFunctionに定位伝達関数 を設定する。)** 116](#_Toc380657318)

[**98.** **harkio\_TransferFunction\_setSepTFs(harkio\_TransferFunctioに分離用伝達関数を設定する)** 117](#_Toc380657319)

[**99.** **harkio\_TransferFunction\_write（伝達関数をzipに出力する）** 118](#_Toc380657320)

[**100.** **harkio\_TransferFunction\_getTFByPos(許容差範囲内のpositionの伝達関数を取得する)** 119](#_Toc380657321)

[**101.** **harkio\_TransferFunction\_getTFById (同じid番号の伝達関数を取得する)** 120](#_Toc380657322)

[102. harkio\_TransferFunction\_getNeighborTFsById (対象IDの近接ID郡の伝達関数を取得する) 121](#_Toc380657323)

[**103.** **readZippedXML(zipファイルからharkio\_XMLに変換する)** 122](#_Toc380657324)

[**104.** **readZippedText (ZIPファイルを読み込む)** 123](#_Toc380657325)

[**105.** **int libharkio3\_TransferFunction\_readlibhakio2 (libharkio2->libhark3変換)** 124](#_Toc380657326)

[**Config (全体設定)** 125](#_Toc380657327)

[**106.** **harkio\_Config\_new (hark\_Config型領域を新規作成する)** 126](#_Toc380657328)

[**107.** **harkio\_Config\_delete (harkio\_Config型領域を削除する)** 127](#_Toc380657329)

[**108.** **harkio\_Config\_getComment (Harkio\_Configからcommentを取得する)** 128](#_Toc380657330)

[**109.** **harkio\_Config\_getSynchronousAverage (Harkio\_ConfigからSynchronousAverageを取得する)** 129](#_Toc380657331)

[**110.** **harkio\_Config\_getTSPpath (Harkio\_ConfigからTSPpathを取得する)** 130](#_Toc380657332)

[**111.** **harkio\_Config\_getTSPoffset (Harkio\_ConfigからTSPoffsetを取得する)** 131](#_Toc380657333)

[**112.** **harkio\_Config\_getPeakSearchFrom (Harkio\_ConfigからPeakSearchFromを取得する)** 132](#_Toc380657334)

[**113.** **harkio\_Config\_getPeakSearchTo (Harkio\_ConfigからPeakSearchToを取得する)** 133](#_Toc380657335)

[**114.** **harkio\_Config\_getNfft (Harkio\_ConfigからNfftを取得する)** 134](#_Toc380657336)

[**115.** **harkio\_Config\_setSynchronousAverage (Harkio\_Config にSynchronousAverageをセットする)** 135](#_Toc380657337)

[**116.** **harkio\_Config\_setTSPpath (Harkio\_ConfigにTSPpathをセットする)** 136](#_Toc380657338)

[**117.** **harkio\_Config\_setTSPoffset (Harkio\_ConfigにTSPoffsetをセットする)** 137](#_Toc380657339)

[**118.** **harkio\_Config\_setPeakSearchFrom (hrkio\_ConfigにPeakSearchFromをセットする)** 138](#_Toc380657340)

[**119.** **harkio\_Config\_setPeakSearchTo (hrkio\_ConfigにPeakSearchToをセットする)** 139](#_Toc380657341)

[**120.** **harkio\_Config\_setNfft (Harkio\_ConfigにNfftをセットする)** 140](#_Toc380657342)

[**121.** **harkio\_Config\_setComment(Harkio\_ConfigにCommentをセットする)** 141](#_Toc380657343)

[**122.** **harkio\_Config\_isEqual（ harkio\_Configの比較）** 142](#_Toc380657344)

[**123.** **harkio\_Config\_prin　（harkio\_Configの内容を出力する）** 143](#_Toc380657345)

[**124.** **harkio\_Config\_copy (harkio\_Configの内容をコピーする)** 144](#_Toc380657346)

[**LOG (ログ出力)** 145](#_Toc380657347)

[**125.** **harkio\_Log\_setLevel (printlevelを設定する)** 146](#_Toc380657348)

[**126.** **harkio\_Log\_printf (ログを表示する)** 147](#_Toc380657349)

[**127.** **void harkio\_Log\_FileOpenFailed (ファイルオープンエラーログを表示する)** 148](#_Toc380657350)

[**128.** **harkio\_Log\_printf (valueが不正の場合のエラーログを表示する)** 149](#_Toc380657351)

[**129.** **harkio\_Log\_NullPointer (pointerがNULLの場合のエラーログを表示する)** 150](#_Toc380657352)

# **構造体一覧表**

## **enum**

|  |  |  |  |
| --- | --- | --- | --- |
| enum | Coordinate | {　Polar,　 Cartesian　} |  |
|  | NeighborAlgorithm | {　Undefined　,　NearestNeighbor　} |  |
|  | TransferType | { loctf , septf } |  |
|  | Calculate | { Calc\_Add , Calc\_Sub , Calc\_Multi , Calc\_Div } |  |
|  | DataType | { INT32 , FLOAT32 , COMPLEX } |  |
|  | UseTagType | { NON\_TAG , USE\_TAG , ID\_TAG } |  |

## **multipleType**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef **union** | 属性 | 名称 | 説明 |
| multipleType; | int | \*intdata; | int型データ |
|  | float | \*floatdata; | float型データ |
|  | complex | \*complexdata | complex型データ |

## **harkio\_Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_Matrix | char | tag[32]; | バージョン：　"1.3" |
|  | char | type[32]; | データタイプ：　int32/float32/complex |
|  | int | dim | Matrixの次元数 |
|  | int | rows; | 行数　　※1 |
|  | int | cols; | 列数　　※2 |
|  | multipleType | data; | データ |

**※１：**伝達関数の場合　チャンネル数がセットされる

※２**：**伝達関数の場合　FFT数がセットされる

## **harkio\_Position**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_Position | int | id; | 通し番号  (-1：positionsに追加時に、最後の番号+1になる) |
|  | enum　Coordinate | sys; | Polar/Cartesian |
|  | float | \*coord; | 座標　 sizeof(float) \* 3  Polar：　[0]azimuth [1]elevation [2]radius  Cartesian：　[0]X [1]Y [2]Z |
|  | char | \*path; | 対応ファイルのパス |
| ※１:channel項目は、  Positionごとにマイクを選択できるようにするため | int | channelsUse | channelsタグ  0:channels useタグ未使用（初期値）  1:channels useタグ使用する場合  2:channel id タグ使用する場合 |
|  | int | numChannels | 下記のchannnel数 |
|  | int | \*channels | 使用するchannel |

## **harkio\_Config**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_Config; | char | \*comment; | コメント |
|  | int | synchronousAverage; | TSP信号の加算回数 |
|  | char | \*TSPpath; | TSP信号ファイルのパス |
|  | int | TSPoffset; | TSPオフセット値 |
|  | int | peakSearchFrom; | 開始サンプル位置（初期値 0） |
|  | int | peakSearchTo; | 終了サンプル位置（初期値 0） |
|  | int | nfft; | FFT 長 |
|  | int | samplingRate | サンプリング周波数 |
|  | int | signalMax | 振幅値 |
|  | int | TSPLength |  |

## **harkio\_Positions**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_Positions | int | size; | 構造体のサイズ |
|  | int | frame; | フレーム番号 |
|  | char | \*type; | tsp/impulse/noise/microphone/soundsource |
|  | harkio\_Position | \*\*pos; | 位置情報  ※id=-1の場合は最終id+1を付加 |

## **harkio\_Neighbors**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_Neighbors | int | size; | サイズ  ※下記posesのsizeを設定  ※new でセットされる |
|  | int | \*ids; | 通し番号群  ※new でセットされる |
|  | int | \*numNeighbors; | 近接ID数  ※Algorithmsでセット |
|  | int | \*\*neighbors; | 近接ID番号  ※Algorithmsでセット |
|  | harkio\_Positions | \*poses; | 位置情報 |
|  | enum　NeighborAlgorithm | alg; | NearestNeighbor |

## **harkio\_XML**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_XML | harkio\_Positions | \*poses; | 位置情報 |
|  | harkio\_Neighbors | \*nbrs; | 近接ID情報 |
|  | harkio\_Config | \*cfg; | 設定値 |

## **harkio\_TransferFunction**

|  |  |  |  |
| --- | --- | --- | --- |
| typedef struct | 属性 | 名称 | 説明 |
| harkio\_TransferFunction | struct　zip | \*handle | zip handle |
|  | harkio\_Positions | \*poses | 位置情報 |
|  | harkio\_Positions | \*mics | マイク位置 |
|  | harkio\_Config | \*cfg; | 設定値 |
|  | harkio\_Neighbors | \*nbrs; | 近接ID情報 |
|  | harkio\_Matrix | \*\*loctfs | 定位用伝達関数 |
|  | harkio\_Matrix | \*\*septfs | 分離用伝達関数 |

# **Matrix**

## harkio\_Matrix\_new　（新規harkio\_Matrix」型構造体作成）

* **機能**

新規に「harkio\_Matrix」型構造体を作成する。

data領域は、引数typeの型で、rows \* cols 分の領域を作成する。

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_new(char \*type, int rows, int cols)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*type | int32/float32/complex |
| IN | int rows | 行数 |
| IN | int cols | 列数 |

* **戻り値**

成功　：　新規 harkio\_Matrix

失敗　：　NULL

* **処理説明**

## harkio\_Matrix\_ones　（すべての要素が 1 の配列を作成）

* **機能**

すべての要素が 1 の配列を作成する

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_ones(char \*type, int rows, int cols)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*type | int32/float32/complex |
| IN | int rows | 行数 |
| IN | int cols | 列数 |

* **戻り値**

成功　：　新規 [**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **処理説明**

## harkio\_Matrix\_ zeros　（すべての要素が 0 の配列を作成）

* **機能**

すべての要素が 0 の配列を作成する

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_zeros(char \*type, int rows, int cols)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*type | int32/float32/complex |
| IN | int rows | 行数 |
| IN | int cols | 列数 |

* **戻り値**

成功　：　新規 [**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **処理説明**

## harkio\_Matrix\_eye　（対角要素が 1 でその他の要素が 0 の m 行 n 列の行列を作成）

* **機能**

単位行列。

対角要素が 1 でその他の要素が 0 の m 行 n 列の行列を作成する

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_eye(char \*type, int size)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*type | int32/float32/complex |
| IN | int size | size（行数と列数は、同じこの値を使用） |
|  |  |  |

* **戻り値**

成功　：　新規 [**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **使用関数**

harkio\_Matrix\_zeros

harkio\_Matrix\_setValueInt32

* **処理説明**

## harkio\_Matrix\_delete　（作成したhark\_Matrix型構造体のメモリを削除（解放））

* **機能**

作成したhark\_Matrix型構造体のメモリを削除（解放）する

* **構文**

**void harkio\_Matrix\_delete(harkio\_Matrix \*\*m)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*\*m |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## harkio\_Matrix\_setValueInt32　（int型の行列の指定箇所（row , col）にvalue値をセットする）

* **機能**

int型の行列の指定箇所（row , col）にvalue値をセットする

* **構文**

**int harkio\_Matrix\_setValueInt32(harkio\_Matrix \*m, int row, int col, int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
|  | int row | 指定行 |
|  | int col | 指定列 |
|  | int value | セットする値 |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

row

harkio\_Matrix \*m

value

col

## harkio\_Matrix\_getValueInt32　（int型の行列の指定箇所から要素を取り出しvalueに設定）

* **機能**

int型の行列mの指定箇所（row , col）から要素を取り出しvalueに設定する

* **構文**

**int harkio\_Matrix\_getValueInt32(harkio\_Matrix \*m, int row, int col, int \*value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
| IN | int row | 指定行 |
| IN | int col | 指定列 |
| IN | int value | セットする値 |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

col

row

harkio\_Matrix \*m

value

## harkio\_Matrix\_setValueFloat32　（float型の行列の指定箇所（row , col）にvalue値をセットする）

* **機能**

float型の行列の指定箇所（row , col）にvalue値をセットする

* **構文**

**int harkio\_Matrix\_setValueFloat32(harkio\_Matrix \*m, int row, int col, float value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
| IN | int row | 指定行 |
| IN | int col | 指定列 |
| IN | int value | セットする値 |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

col

row

harkio\_Matrix \*m

value

## harkio\_Matrix\_getValueFloat32　（float型の行列の指定箇所から要素を取り出しvalueにセット）

* **機能**

float型の行列mの指定箇所から要素を取り出しvalueにセットする

* **構文**

int harkio\_Matrix\_getValueFloat32(harkio\_Matrix \*m, int row, int col, float \*value)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
| IN | int row | 指定行 |
| IN | int col | 指定列 |
| OUT | int value | 取得した値 |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

row

harkio\_Matrix \*m

value

col

## harkio\_Matrix\_setValueComplex（float型の行列にvalue値をセットする）

* **機能**

complex型の行列の指定箇所（row , col）にvalue値をセットする

* **構文**

**int harkio\_Matrix\_setValueComplex32(harkio\_Matrix \*m,**

**int row,**

**int col,**

**complex value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
| IN | int row | 指定行 |
| IN | int col | 指定列 |
| IN | complex value | セットする値 |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

col

row

harkio\_Matrix \*m

value

## harkio\_Matrix\_getValueComplex32　（complex型の行列から要素を取り出しvalueにセット）

* **機能**

complex型の行列mの指定箇所から要素を取り出しvalueにセットする

* **構文**

int harkio\_Matrix\_getValueComplex32(harkio\_Matrix \*m,

int row,

int col,

float \*value)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*m | hark\_Matrix型ポインタ |
| IN | int row | 指定行 |
| IN | int col | 指定列 |
| OUT | complex value | 取得した値 |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

row

harkio\_Matrix \*m

value

col

## harkio\_Matrix\_getValues　（指定範囲の要素を取得する）

* **機能**

指定範囲の要素を取得する

* **構文**

harkio\_Matrix\* harkio\_Matrix\_getValues(harkio\_Matrix \*matrix,

introwFrom ,

int rowTo,

int colFrom,

int colTo)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*matrix | 取り出し元 |
| IN | int rowFrom | 取り出し開始行 |
| IN | int rowTo | 取り出し終了行 |
| IN | int colFrom | 取り出し開始列 |
| IN | int colTo | 取り出し終了列 |

* **戻り値**
* 成功　：　[**harkio\_Matrix\***](#_harkio_Matrix)
* 失敗　：　NULL
* **処理説明**

colFrom

rowFrom

harkio\_Matrix \*matrix

harkio\_Matrix \*new

colTo

rowTo

## harkio\_Matrix\_setValues　（Matrix型(複数要素)をセットする）

* **機能**

Matrix型(複数要素)をセットする

* **構文**

int harkio\_Matrix\_setValues(harkio\_Matrix \*matrix\_src,

harkio\_Matrix \*matrix\_dest,

int row,

int col)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*matrix\_src | セット先 |
| IN | harkio\_Matrix \*matrix\_dest | セットするデータ |
| IN | int row | セット開始行 |
| IN | int col | セット開始列 |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

col

row

harkio\_Matrix \*dest

harkio\_Matrix \*src

## harkio\_Matrix\_add　（行列aと行列bの各要素の和）

* **機能**

行列aと行列bの各要素の和

* **構文**

harkio\_Matrix\* harkio\_Matrix\_add(harkio\_Matrix \*a, harkio\_Matrix \*b)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*a | hark\_Matrix型ポインタ |
| IN | harkio\_Matrix \*b | hark\_Matrix型ポインタ |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　harkio\_Matrix\*

失敗　：　NULL

* **備考**

行列aと行列bの行数、列数とtypeは、同一のこと

* **処理説明**

A1+B1=C

A2+B2=D

C

D

B1

B2

B3

B5

B4

B6

A1

A2

harkio\_Matrix \*

harkio\_Matrix \*b

harkio\_Matrix \*a

## harkio\_Matrix\_sub　（行列aと行列bの各要素の差）

* **機能**

行列aと行列bの各要素の差

* **構文**

harkio\_Matrix\* harkio\_Matrix\_sub(harkio\_Matrix \*a, harkio\_Matrix \*b)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*a | hark\_Matrix型ポインタ |
| IN | harkio\_Matrix \*b | hark\_Matrix型ポインタ |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　harkio\_Matrix\*

失敗　：　NULL

* **備考**
* **行列aと行列bの行数、列数とtypeは、同一のこと**
* **処理説明**

harkio\_Matrix \*b

C

D

B1

B2

B3

B5

B4

B6

A1

A2

harkio\_Matrix \*a

A1-B1=C

A2-B2=D

harkio\_Matrix \*

## harkio\_Matrix\_mul　（行列の積）

* **機能**

行列の積

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_mul( harkio\_Matrix \*a , harkio\_Matrix \*b)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | [harkio\_Matrix \*a](#_harkio_Matrix) | hark\_Matrix型構造体のポインタ |
| IN | [harkio\_Matrix \*b](#_harkio_Matrix) | hark\_Matrix型構造体のポインタ |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　[**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **処理説明**

A1

A2

B1

B2

B3

B4

C

D

A1\*B1+A2\*B3 = C

A1\*B2+A2\*B4 = D

hark\_Matrix\*

hark\_Matrix\*

hark\_Matrix\*

hark\_Matrix\*

harkatrix\*

## harkio\_Matrix\_emul　（行列の各要素ごとの積）

* **機能**

行列の各要素ごとの積

* **構文**

harkio\_Matrix\*　　harkio\_Matrix\_emul(**harkio\_Matrix \*a , harkio\_Matrix \*b**)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | [harkio\_Matrix \*a](#_harkio_Matrix) | hark\_Matrix型構造体のポインタ |
| IN | [harkio\_Matrix \*b](#_harkio_Matrix) | hark\_Matrix型構造体のポインタ |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　[**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **備考**
* **行列aと行列bの行数、列数とtypeは、同一のこと**
* **処理説明**

harkio\_Matrix \*

harkio\_Matrix \*b

C

D

B1

B2

B3

B5

B4

B6

A1

A2

harkio\_Matrix \*a

A1\*B1=C

A2\*B2=D

## harkio\_Matrix\_sum　（行列の合計）

* **機能**

行列の合計

* **構文**

**harkio\_Matrix\* harkio\_Matrix\_sum( harkio\_Matrix \*\*matrix , int num )**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | [harkio\_Matrix \*\*matrix](#_harkio_Matrix) | hark\_Matrix型構造体のポインタ |
| IN | int num | Matrixの総数 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　[**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

* **処理説明**

## harkio\_Matrix\_fromFile　（ファイルの内容をharkio\_Matrix型構造体にセットする）

* **機能**

ファイルの内容をharkio\_Matrix型構造体にセットする

* **構文**

harkio\_Matrix\* harkio\_Matrix\_fromFile(char \*path)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*path | 読み込むファイルのパス |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　[**harkio\_Matrix\***](#_harkio_Matrix)

失敗　：　NULL

t

* **処理説明**

## harkio\_Matrix\_print　（harkio\_Matrix構造体の内容をすべて表示する）

* **機能**

harkio\_Matrix構造体の内容をすべて表示する

* **構文**

**void harkio\_Matrix\_print(harkio\_Matrix \*m)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | [**harkio\_Matrix\***](#_harkio_Matrix) | 出力対象 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **処理説明**

## harkio\_Matrix\_write　（harkio\_Matrix型構造体の内容をファイルに書き出す）

* **機能**

harkio\_Matrix型構造体の内容をファイルに書き出す

* **構文**

int harkio\_Matrix\_write(harkio\_Matrix \*m, char \*path)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | [**harkio\_Matrix\***](#_harkio_Matrix) | 書き出す内容 |
|  | char \*path | 書き込み先ファイルのパス |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

Harkio\_Matrix \*m

char \*path

## harkio\_Matrix\_isEqual　（行列比較）

* **機能**

入力のhark\_Matrix構造体の内容が同じ場合は EXIT\_SUCCESS を返し、違えば EXIT\_FAILUREを返す

* **構文**

**int harkio\_Matrix\_isEqual(harkio\_Matrix \*a, harkio\_Matrix \*b)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Matrix \*a | hark\_Matrix型ポインタ |
| IN | harkio\_Matrix \*b | hark\_Matrix型ポインタ |
|  |  |  |
|  |  |  |

* **戻り値**

内容が同一　：　EXIT\_SUCCESS

内容が異なる　：　EXIT\_FAILURE

* **処理説明**

harkio\_Matrix \*b

比較

harkio\_Matrix \*a

## harkio\_Matrix\_size　（行数と列数を、size配列にセットする）

* **機能**

hark\_Matrix型構造体でセットしている、行数と列数を、size配列にセットする

* **構文**

size\_t harkio\_Matrix\_size(harkio\_Matrix \*m, int \*size)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | int \*size | size[0] ← rows  size[1] ← cols |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

hark\_Matrixのサイズの合計

**sizeof(tag)+sizeof(type)+sizeof(dim)+ sizeof(rows)+sizeof(cols)+sizeof(type size)+rows\*cols**

* **処理説明**

## harkio\_Matrix\_ fromZip（zipファイルからharkio\_Matrixに変換する）

* **機能**

zipファイルから入力で指定されたpathのファイルを抽出して、harkio\_Matrixに変換する

* **構文**

harkio\_Matrix\* harkio\_Matrix\_fromZip(struct zip \*handle, char \*path)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | struct zip \*handle | zipファイルのファイルポインタ |
| IN | char \*path | harkio\_Matrixに変換するファイルのパス |
|  |  |  |
|  |  |  |

* **戻り値**

**hark\_Matrix\***

* **処理説明**

zip

localization

TransferFunction

tf00000.mat

hark\_Matrix

指定

path

# **XML**

## harkio\_XML\_new　（新規harkio\_XML型構造体作成）

* **機能**

新規に「harkio\_XML」型構造体を作成する。

* **構文**

**harkio\_XML\* harkio\_XML\_new(void)**

* **パラメータ**

なし

* **戻り値**

成功　：　新規 [**harkio\_XML\***](#_harkio_XML)

失敗　：　NULL

* **処理説明**

## harkio\_XML\_newFromXML（XMLファイルの内容を「harkio\_XML」型構造体に設定）

* **機能**

XMLファイルを読み込んで「harkio\_XML」型構造体に設定する。

* **構文**

**harkio\_XML \* harkio\_XML\_newFromXML(char \*xmlpath, char \*dtdpath)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | char \*xmlpath | ファイルのパス |
|  | Char\* dtdpath |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　新規 [**harkio\_XML\***](#_harkio_XML)

失敗　：　NULL

* **処理説明**

## harkio\_XML\_getTag　(tag(version)を取得する)

* **機能**

tag(version)を取得する

* **構文**

**void harkio\_XML\_getTag (harkio\_XML \*xml , char\* value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

## harkio\_XML\_getType　(typeを取得する)

* **機能**

入力情報から、Typeを取得する

* **構文**

**void harkio\_XML\_getConfig (harkio\_XML \*xml , char \* valu)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

## harkio\_XML\_getRow　(rowを取得する)

* **機能**

入力情報から、rowを取得する

* **構文**

**int harkio\_XML\_getConfig (harkio\_XML \*xml )**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

row

## harkio\_XML\_getCol　(colを取得する)

* **機能**

入力情報から、colを取得する

* **構文**

**int harkio\_XML\_getCol (harkio\_XML \*xml )**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**col**

## harkio\_XML\_newFromData　（入力パラメータから、新規に「harkio\_XML」型構造体を作成）

* **機能**

入力パラメータ（位置情報, 近接ID, 設定値）から、新規に「harkio\_XML」型構造体を作成する。

* **構文**

**harkio\_Matrix\* harkio\_XML\_newFromData(harkio\_Positions \*ps,**

**harkio\_Neighbors \*nbrs,**

**harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | harkio\_Positions \*ps | 位置情報 |
|  | harkio\_Neighbors \*nbrs | 近接ID |
|  | harkio\_Config \*cfg s | 設定値 |

* **戻り値**

成功　：　新規 [**harkio\_XML\***](#_harkio_XML)

失敗　：　NULL

* **処理説明**

## harkio\_XML\_delete　(Harkio\_XML型構造体の領域を開放する)

* **機能**

Harkio\_XML型構造体の領域を開放する

* **構文**

**void harkio\_XML\_delete(harkio\_XML \*\*xml)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*\*xml | 開放する領域 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **処理説明**

## harkio\_XML\_getNeighbors　（近接ID情報を取得する）

* **機能**

入力情報から、近接ID情報を取得する

* **構文**

**harkio\_Neighbors \* harkio\_XML\_getNeighbors(harkio\_XML \*xml)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力元データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　新規 [**harkio\_Neighbors \***](#_harkio_Neighbors)

失敗　：　NULL

## harkio\_XML\_getPositions　(位置情報を取得する)

* **機能**

入力情報から、位置情報を取得する

* **構文**

**harkio\_Position \* harkio\_XML\_getPositions (harkio\_XML \*xml)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　新規 [**harkio\_Position\***](#_harkio_Position)

失敗　：　NULL

## harkio\_XML\_setPositions　(位置情報をセットする)

* **機能**

入力情報を、harkio\_Positions領域にセットする

* **構文**

**int harkio\_XML\_setPositions (harkio\_XML \*xml , harkio\_Positions \*pos)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
| OUT | harkio\_Positions \*pos | 出力先 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

## harkio\_XML\_getConfig　(位置情報を取得する)

* **機能**

入力情報から、位置情報を取得する

* **構文**

**harkio\_Config \* harkio\_XML\_getConfig (harkio\_XML \*xml)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　新規 harkio\_Config\*

失敗　：　NULL

## harkio\_XML\_write　(XML形式で出力する)

* **機能**

XML形式で出力する。

hark\_XML構造体からXML形式に変換して,streamに出力する

* **構文**

**int harkio\_XML\_write(harkio\_XML \*xml, FILE \*fstream)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力元データ |
| OUT | FILE \*fstream | 出力ファイルストリーム |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

## harkio\_XML\_writeBuffer　(XML形式に変換してバッファに出力する)

* **機能**

入力データを、XML形式に変換して、バッファに出力する。

* **構文**

char\* harkio\_XML\_writeBuffer(harkio\_XML \*xml)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力元データ |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　Char\*(新規buffer領域)

失敗　：　NULL

## harkio\_XML\_internal\_write　(XML形式に出力する。)

* **機能**

XML形式に出力する。

* **構文**

int harkio\_XML\_internal\_write(harkio\_XML \*xml, FILE \*fstream, char \*cstream)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_XML \*xml | 入力データ |
| OUT | FILE \*fstream | 出力ストリーム |
| OUT | char \*cstream | 入力データ |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**

## harkio\_XML\_internal\_processNode　（読み込んだデータを、「harkio\_XML」型構造体に変換する）

* **機能**

読み込んだデータを、「harkio\_XML」型構造体に変換する。

* **構文**

**int harkio\_XML\_internal\_processNode(xmlTextReaderPtr reader, harkio\_XML \*xml)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| INT | xmlTextReaderPtr reader | 入力データ |
| OUT | harkio\_XML \*xml | 出力 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

## harkio\_XML\_internal\_strcount　(文字列の総数を取得する)

* **機能**

入力の文字列から、入力のKeyをカウントし、総数を返す

* **構文**

int harkio\_XML\_internal\_strcount(const char \*str, char key)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | const char \*str | 文字列 |
| IN | char key | 検索key |
|  |  |  |

* **戻り値**

Keyの総数

## pushStream　（Streamに出力する）

* **機能**

Streamに出力する

* **構文**

void pushStream(FILE \*fstream,

char \*cstream,

char \*string)

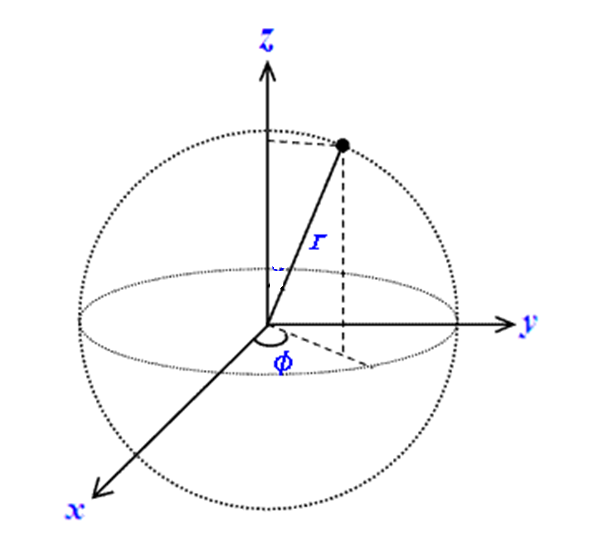
* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | FILE \*fstream | 格納先ファイルstream |
| OUT | char \*cstream | 格納先領域 |
| IN | char \*string | 文字列 |
|  |  |  |

* **戻り値**

なし

# **Position (位置)**

****

(x,y,z)

(r,θ,φ)

√x^2 + y^2

polar ⇒ Cartesian変換

x = r\*cos(θ)\*cos(φ)

y = r\*cos(θ)\*sin(φ)

z = r\*sin(θ)

Azimuth(φ)

Elevation(θ)

## **harkio\_Position\* harkio\_Position\_new（新規に「harkio\_Position」型構造体を作成する）**

* **機能**

新規に「harkio\_Position」型構造体を作成する。

* **構文**

**harkio\_Position\* harkio\_Position\_new(int id,**

**enum Coordinate sys,**

**float \*coord,**

**char \*path)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | int id | id |
| IN | enum Coordinate sys | Polar / Cartesian |
| IN | float \*coord | 座標 |
| IN | char \*path | 対応するファイルのパス  TSPファイル/Micファイルのパス  ファイルが無い場合：NULL |

* **戻り値**

**harkio\_Position\***

* **処理説明**

## **harkio\_Position\_delet (「harkio\_Position」型構造体を削除する)**

* **機能**

「harkio\_Position」型構造体を削除する。

* **構文**

void harkio\_Position\_delete(harkio\_Position \*\*p)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*\*p | 削除対象領域 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **処理説明**

## **float\* harkio\_Position\_getPosition（座標を取得する）**

* **機能**

座標を取得する

* **構文**

float\* harkio\_Position\_getPosition(harkio\_Position\* pos)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position\* pos | 取得対象 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

harkio\_Position->coord

* **処理説明**

## **harkio\_Position\_setPosition（Positionをセットする）**

* **機能**

座標系および座標を設定する

* **構文**

int harkio\_Position\_setPosition(harkio\_Position\* pos,

enum Coordinate sys,

float \*coord)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Position\* pos | 設定する領域 |
| IN | enum Coordinate sys | 設定する座標系 |
| IN | float \*coord | 設定する座標 |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**
* **処理説明**

## **harkio\_Position\_setChannels（channelsをセットする）**

* **機能**

channnelsを設定する

* **構文**

int harkio\_Position\_setPosition(harkio\_Position\* pos,

int \*channnels)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Position\* pos | 設定する領域 |
| IN | int \*channels | 設定するchannelsのアドレス |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**
* **処理説明**

## **harkio\_Position\_setChannels（channelsをセットする）**

* **機能**

channnelsを設定する

* **構文**

int harkio\_Position\_setChannels(harkio\_Position\* pos,

int \*channnels)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Position\* pos | 設定する領域 |
| IN | int \*channels | 設定するchannelsのアドレス |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**
* **処理説明**

## **harkio\_Position\_setNumChannels（numChannelsをセットする）**

* **機能**

numChannnelsを設定する

* **構文**

int harkio\_Position\_setNumChannels(harkio\_Position\* pos,

int numchannnels)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Position\* pos | 設定する領域 |
| IN | int numchannels | channelsの総数 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**
* **処理説明**

## **harkio\_Position\_setChannelsUse（ChannelsUseをセットする）**

* **機能**

channnelsUseを設定する

* **構文**

int harkio\_Position\_setNumChannels(harkio\_Position\* pos,

int channnelsUse)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Position\* pos | 設定する領域 |
| IN | int channelsUse | channels useのステータス  NON\_TAG：USEタグ不使用  USE\_TAG：USEタグ使用  ID\_TAG：IDタグ使用 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **使用関数**
* **処理説明**

<channels use="1;2;4;"> ⇒　使用しない NON\_TAG / 使用する USE\_TAG　をセット

<channel id="0" use="1;3;5;"/> ⇒使用する ID\_TAGをセット

</channels>

## **harkio\_Position\_getID (idを取得する)**

* **機能**

idを取得する

* **構文**

int harkio\_Position\_getID(harkio\_Position \*pos)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*pos | harkio\_Position |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　id

失敗　: -1

* **処理説明**

## **harkio\_Position\_getSys (sys(coordinate)を取得する)**

* **機能**

sysを取得する

* **構文**

int harkio\_Position\_getSys(harkio\_Position \*pos)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*pos | harkio\_Position |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　sys

**失敗　：-1**

* **処理説明**

## **harkio\_Position\_getPath (Pathを取得する)**

* **機能**

Pathを取得する

* **構文**

void harkio\_Position\_getCoord(harkio\_Position \*pos , char \*value)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*pos | harkio\_Position |
| out | float \*value | path |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Position\_print（Positionを表示する）**

* **機能**

引数Positionを表示する

* **構文**

**void harkio\_Position\_print(harkio\_Position \*pos)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position\* pos | 表示対象 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **処理説明**

## **harkio\_Position\_isEqual（ hark\_positionの比較）**

* **機能**

入力されたhark\_positionの比較を行う

* **構文**

**int harkio\_Position\_isEqual(harkio\_Position \*a, harkio\_Position \*b)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*a | 比較対象 |
| IN | harkio\_Position \*b | 比較対象 |
|  |  |  |
|  |  |  |

* **戻り値**

０　：　不一致

1　：　一致

* **処理説明**

## **harkio\_Position\_copy（オブジェクトのコピーを行う）**

* **機能**

入力されたオブジェクトのコピーを行う

* **構文**

harkio\_Position\* harkio\_Position\_copy(harkio\_Position \*p)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*p | コピー元 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Position\***](#_harkio_Position)　コピー先

* **処理説明**

## **harkio\_Position\_Convert（座標変換する）**

* **機能**

座標をpolarからcartesian

または、

cartesianからpolarに変換する

* **構文**

**int harkio\_Position\_Convert(harkio\_Position \*pos, float \*\*coord , int type)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*a | 変換対象 |
| out | float \*\*coord | 変換後のXYZ座標がセットされる領域　float coord[3] |
| IN | int type | 変換する座標の種別  Polar　　　　 Cartesian -> Polarに変換  Cartesian　 Polar -> Cartesianに変換 |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

x = r \* cos(theta) \* cos(phi);

y = r \* cos (theta) \* sin(phi);

z = r \* sin(theta);

//直交座標(x,y,z)を極座標 に変換する

θ = atan(Z/SQRT( x^2 + Y^2))

φ = atan( Y/X)

r = SQRT( x^2 + Y^2 + Z^2)

## **harkio\_Position\_convToRadian (角度(degree)からラジアン(radian)に変換する)**

* **機能**

角度(degree)からラジアン(radian)に変換

* **構文**

float harkio\_Position\_convToRadian(float degree)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | float degree | 変換元（degree値） |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　変換後の値(radian値)

失敗　：　なし

* **処理説明**

## **harkio\_Position\_convToRadian (ラジアン(radian)から角度(degree)に変換する)**

* **機能**

ラジアン(radian)から角度(degree)に変換

* **構文**

float harkio\_Position\_convToDegree (float radian)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | float radian | 変換元（radian値） |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　変換後の値(degree値)

失敗　：　なし

* **処理説明**

# **Positions (位置の集合)**

## harkio\_Positions\_new（新規harkio\_Positions型構造体作成）

* **機能**

新規で、harkio\_Positions型を作成する

* **構文**

**harkio\_Positions\* harkio\_Positions\_new(char \*type,**

**int frame,**

**int size,**

**harkio\_Position \*\*pos)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*type | "tsp", "impulse", "noise", "microphone", "soundsource" |
| IN | int frame | フレーム番号 |
| IN | int size | サイズ(下記harkio\_position群の総数) |
| IN | harkio\_Position \*\*pos | positio群 |

* **戻り値**

成功　：　新規 [**harkio\_Positions\***](#_harkio_Positions)

失敗　：　NULL

* **処理説明**

## harkio\_Positions\_delete(harkio\_Positions型領域を削除)

* **機能**

作成したharkio\_Positions型領域を削除する

* **構文**

**void harkio\_Positions\_delete(harkio\_Positions \*\*pos)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*\*pos | 削除する領域 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **処理説明**

harkio\_Position\_delete((\*pos)->pos + i);

free( (\*pos)->type);

## **harkio\_Positions\_getById (入力したIDと同じIDのharkio\_Positionを返す)**

* **機能**

入力したharkio\_Positionsから入力したIDと同じIDのharkio\_Positionを返す

* **構文**

harkio\_Position\* harkio\_Positions\_getById(harkio\_Positions \*poses, int id)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | 検索対象 |
| IN | int id | 検索id |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　[harkio\_Position\*](#_harkio_Position)

失敗　：　NULL

* **処理説明**

## **harkio\_Positions\_getByCoord (入力パラメータからHark\_positionを取得する)**

* **機能**

入力パラメータからHark\_positionを取得する

* **構文**

harkio\_Position\* harkio\_Positions\_getByCoord(harkio\_Positions \*poses,

enum Coordinate coordinate,

float \*coord,

float tolerance)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
| IN | num Coordinate coordinate | 座標系 |
| IN | float \*coord | 座標  ※Cartesian座標を入力する  ※polar座標のpositionは、PG内部でCartesianに変換される。 |
| IN | float tolerance | 許容範囲 |

* **戻り値**

成功　：　[harkio\_Position\*](#_harkio_Position)

失敗　：　NULL

* **処理説明**

## **harkio\_Positions\_getType (Typeを取得する)**

* **機能**

Typeを取得する

* **構文**

char\* harkio\_Positions\_getType(harkio\_Positions \*poses)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　char

* **処理説明**

## **harkio\_Positions\_getSize (sizeを取得する)**

* **機能**

sizeを取得する

* **構文**

int harkio\_Positions\_getSize(harkio\_Positions \*poses)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　size

失敗　：　-1

* **処理説明**

## **harkio\_Positions\_getFrame (frameを取得する)**

* **機能**

frameを取得する

* **構文**

int harkio\_Positions\_getFrame(harkio\_Positions \*poses)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　frame

失敗　：　-1

* **処理説明**

## **harkio\_Positions\_readFiles (ファイルから読み込む)**

* **機能**

ファイルから読み込む

* **構文**

int harkio\_Positions\_readFiles(harkio\_Positions \*poses)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_Positions\_append（Hark\_positonsにhark\_positionを追加する）**

* **機能**

Hark\_positonsにhark\_positionを追加する

* **構文**

void harkio\_Positions\_append( harkio\_Positions \*poses, harkio\_Position \*pos)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | 追加先領域 |
| IN | harkio\_Position \*pos | 追加position |

* **戻り値**

**なし**

* **処理説明**

hark\_position分、領域を拡張して、hark\_positionを追加する

realloc

positions

append

## **harkio\_Positions\_removeById (削除対象id のpositionを削除する)**

* **機能**

Hark\_Positinsから削除対象id のpositionを削除する

* **構文**

**int harkio\_Positions\_removeById(harkio\_Positions \*poses, int id)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | 削除対象positions |
| IN | int id | 削除対象id |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

削除対象position

削除後、size-1に変更

新しいposition

古いposition

新しいpositionをセットする

古いpositionは削除する

新しいpositionをセットする

あたらしいpositionをセット

|  |  |  |
| --- | --- | --- |
| 属性 | 名称 | 説明 |
| int | size; | 構造体のサイズ |
| int | frame; | フレーム番号 |
| char | \*type; | type |
| harkio\_Position | \*\*pos; | 位置情報 |

## **harkio\_Positions\_removeByCoord (削除対象座標のpositionを削除する)**

* **機能**

Hark\_Positinsから削除対象id のpositionを削除する

* **構文**

int harkio\_Positions\_removeByCoord(**harkio\_Positions \*poses,**

**enum Coordinate coordinate,**

**float \*coord,**

**float tolerance**)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*poses | harkio\_Positions |
|  | enum Coordinate coordinate | 座標系 |
|  | float\* coord | 削除対象座標 |
|  | float\* tolerance | 範囲 |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **Positions\* harkio\_Positions\_copy（引数のhark\_Positinsのコピーを作成する）**

* **機能**

引数のhark\_Positinsのコピーを作成する

* **構文**

**harkio\_Positions\* harkio\_Positions\_copy(harkio\_Positions \*p1)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*p1 | コピー元 |

* **戻り値**

[**harkio\_Positions\***](#_harkio_Positions)

* **処理説明**

## **harkio\_Positions\_isEqual (Positionを比較する)**

* **機能**

引数のPositionsを比較する

* **構文**

int harkio\_Positions\_isEqual(harkio\_Positions \*a, harkio\_Positions \*b)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*a | 比較対象 |
| IN | harkio\_Positions \*b | 比較対象 |

* **戻り値**

０　：　不一致

1　：　一致

* **処理説明**

## **harkio\_Positions\_print (hark\_positionsの表示をする)**

* **機能**

hark\_positionsの表示をする。

* **構文**

**void harkio\_Positions\_print(harkio\_Positions \*p)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*p |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

# **Neighbors (隣接位置)**

## **harkio\_Neighbors\_new (新規で、harkio\_Neighbors型を作成)**

* **機能**

新規で、harkio\_Neighbors型を作成する

パラメータのharkio\_PositionsがNULLの場合は、初期値をセットして作成する

* **構文**

**harkio\_Neighbors\* harkio\_Neighbors\_new(harkio\_Positions \*poses)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Positions \*\*pos | セットするpositions |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Neighbors\***](#_harkio_Neighbors)

* **処理説明**

## **harkio\_Neighbors\_delete (harkio\_Neighbors型領域を削除)**

* **機能**

作成したharkio\_Neighbors型領域を削除する

* **構文**

void harkio\_Neighbors\_delete(harkio\_Neighbors \*\*nbrs)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*\*nbrs |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

**なし**

* **使用関数**

**free**

* **処理説明**

## **harkio\_Neighbors\_calcNeighbors (アルゴリズムで取得した結果を入力パラメータに設定)**

* **機能**

指定したアルゴリズムで取得した結果を入力パラメータに設定する

* **構文**

int harkio\_Neighbors\_calcNeighbors(harkio\_Neighbors \*nbrs,

enum NeighborAlgorithm alg, ...)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| out | harkio\_Neighbors \*nbrs | 出力先領域 |
| IN | enum NeighborAlgorithm alg | Undefined / NearestNeighbor |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_Neighbors\_append　（Hark\_positonsにhark\_positionを追加する）**

* **機能**

入力パラメータのIDと同じ、Hark\_positonsにhark\_positionを追加する

すでに、存在するIDの場合は、numNeighbors ,neighborsを追加

存在しないIDの場合は、ids, numNeighbors ,neighborsを追加

* **構文**

int harkio\_Neighbors\_append(harkio\_Neighbors \*nbrs,

int id,

int numNeighbors,

int \*neighbors)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Neighbors \*nbrs | 追加先 |
| IN | int id | 追加するid |
| IN | int numNeighbors | 追加するnumNeighbors |
| IN | int \*neighbors | 追加するneighbors |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_Neighbors\_Algorithms\_NearestNeighbor (NearestNeighboを用いて設定する)**

* **機能**

NearestNeighborアルゴリズムをもちいて、harkio\_Neighborsを設定する

* **構文**

**int harkio\_Neighbors\_Algorithms\_NearestNeighbor(harkio\_Neighbors \*nbrs,**

**int n,**

**float thresh)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Neighbors \*nbrs | セットするharkio\_Neighbors |
| IN | int n | 近接数 |
| IN | float thresh | threshold |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_Neighbors \_getSize(sizeを取得する)**

* **機能**

Hark\_Neighborsからsizeを取得する

* **構文**

**int harkio\_Neightbors\_getSize(harkio\_ Neightbors \*nbs )**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*nbs | 取得するharkio\_Neighbors |
|  |  |  |

* **戻り値**

成功　：**size**

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_Neighbors \_getNeighborAlgorithm(algを取得する)**

* **機能**

Hark\_Neighborsからalgを取得する

* **構文**

**int harkio\_Neightbors\_getNeighborAlgorithm(harkio\_ Neightbors \*nbs )**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*nbs | 取得するharkio\_Neighbors |
|  |  |  |

* **戻り値**

成功　：**alg**

失敗　：　EXIT\_FAILURE

## **euclidDistance**

* **機能**
* **構文**

**float euclidDistance(harkio\_Position \*pos\_a, harkio\_Position \*pos\_b, int length)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*pos\_a | harkio\_Position |
| IN | harkio\_Position \*pos\_b | harkio\_Position |
| IN | int length | 次元数: ※３固定 |
|  |  |  |

* **戻り値**

**float**

* **使用関数**
* **処理説明**

## **harkio\_Neighbors\_print (harkio\_Neighborsを表示する)**

* **機能**

harkio\_Neighborsを表示する

* **構文**

void harkio\_Neighbors\_print(harkio\_Neighbors \*nbrs)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*nbrs |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Neighbors\_isEqual (harkio\_Neighborsを比較する)**

* **機能**

引数のharkio\_Neighborsを比較する

* **構文**

int harkio\_Neighbors\_isEqual(harkio\_Neighbors \*a, harkio\_Neighbors \*b)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*a | 比較対象 |
| IN | harkio\_Neighbors \*b | 比較対象 |
|  |  |  |
|  |  |  |

* **戻り値**

０　：　不一致

1　：　一致

* **使用関数**
* **処理説明**

## **harkio\_Neighbors\_copy　(harkio\_Neighborsをコピー)**

* **機能**

引数のharkio\_Neighborsをコピーする

* **構文**

**harkio\_Neighbors\* harkio\_Neighbors\_copy(harkio\_Neighbors \*n1)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*n1 | コピー元 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Neighbors\***](#_harkio_Neighbors)

* **処理説明**

# **TransferFunction (伝達関数操作)**

## harkio\_TransferFunction\_new (harkio\_TransferFunction型領域を新規作成する)

* **機能**

新規で、harkio\_TransferFunction型領域を作成する

* **構文**

harkio\_TransferFunction\* harkio\_TransferFunction\_new(void)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | なし |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[harkio\_TransferFunction\*](#_harkio_TransferFunction)

* **処理説明**

## **harkio\_TransferFunction\_fromData　（）**

* **機能**

新規で、harkio\_TransferFunction型領域を作成する

* **構文**

harkio\_TransferFunction\*

harkio\_TransferFunction\_fromData(harkio\_Positions \*poses,

harkio\_Positions \*mics,

harkio\_Config \*cfg,

harkio\_Neighbors \*nbrs)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[harkio\_TransferFunction\*](#_harkio_TransferFunction)

* **処理説明**

## **harkio\_Neighbors\_delete　（harkio\_Neighbors型領域を削除）**

* **機能**

作成したharkio\_Neighbors型領域を削除する

* **構文**

**void harkio\_Neighbors\_delete(harkio\_Neighbors \*\*nbrs)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Neighbors \*\*nbrs |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_TransferFunction\_fromFile　（ZIPファイルから、伝達関数を読み込む）**

* **機能**

ZIPファイルから、伝達関数を読み込む

* **構文**

harkio\_TransferFunction\* harkio\_TransferFunction\_fromFile(char \*path)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*path | Zipファイルのパス |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_TransferFunction\_getConfig (harkio\_TransferFunctionからHark\_Configを取得する)**

* **機能**

harkio\_TransferFunctionからHark\_Configを取得する

* **構文**

harkio\_Config\* harkio\_TransferFunction\_getConfig(harkio\_TransferFunction \*tf)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*path | Zipファイルのパス |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[harkio\_Config\*](#_harkio_Config)

* **処理説明**

## **harkio\_TransferFunction\_getNeighbors (harkio\_Neighborsを取り出す)**

* **機能**

harkio\_TransferFunctionからharkio\_Neighborsを取り出す

* **構文**

harkio\_Neighbors\*

harkio\_TransferFunction\_getNeighbors(harkio\_TransferFunction \*tf)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf | 伝達関数 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Neighbors\***](#_harkio_Neighbors)

* **処理説明**

## **harkio\_TransferFunction\_getMicrophones (harkio\_TransferFunctionからmic座標を取り出す)**

* **機能**

harkio\_TransferFunctionからmic座標を取り出す

* **構文**

**harkio\_Positions\* harkio\_TransferFunction\_getMicrophones(harkio\_TransferFunction \*tf)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Positions\***](#_harkio_Positions)

* **処理説明**

## **harkio\_TransferFunction\_getPositions (harkio\_TransferFunctionから位置情報を取り出す)**

* **機能**

harkio\_TransferFunctionから位置情報を取り出す

* **構文**

**harkio\_Positions\***

**harkio\_TransferFunction\_getPositions (harkio\_TransferFunction \*tf)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf | 入力 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Positions\***](#_harkio_Positions)

* **処理説明**

## **harkio\_TransferFunction\_setConfig (harkio\_TransferFunctionにharkio\_Configをセットする)**

* **機能**

harkio\_TransferFunctionにharkio\_Configをセットする

* **構文**

**int harkio\_TransferFunction\_setConfig(harkio\_TransferFunction \*tf,**

**harkio\_Config \* value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_setNeighbors (harkio\_Neighborsをセットする)**

* **機能**

harkio\_TransferFunctionにharkio\_Neighborsをセットする

* **構文**

**int　harkio\_TransferFunction\_setNeighbors(harkio\_TransferFunction \*tf,**

**harkio\_Neighbors \*value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
| IN | harkio\_Positions \*value |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功 ：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_setMicrophones (harkio\_TransferFunctionにマイク位置をセットする)**

* **機能**

harkio\_TransferFunctionにマイク位置をセットする

* **構文**

**int harkio\_TransferFunction\_setMicrophones (harkio\_TransferFunction \*tf,**

**harkio\_Positions \* value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_TransferFunction \*tf |  |
| IN | harkio\_Positions \*value |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功 ：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_setPositions (harkio\_TransferFunctionに、harkio\_Positionsを設定する)**

* **機能**

harkio\_TransferFunctionのposesに、harkio\_Positionsを設定する。

* **構文**

**int harkio\_TransferFunction\_setPositions(harkio\_TransferFunction \*tf,**

**harkio\_Positions \*value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
| IN | harkio\_Positions \*value | 設定する値 |
|  |  |  |
|  |  |  |

* **戻り値**

成功 ：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_setLocTFs (harkio\_TransferFunctionに定位伝達関数 を設定する。)**

* **機能**

harkio\_TransferFunctionのloctfsに、harkio\_Matrix\*\*（定位伝達関数） を設定する。

* **構文**

**int harkio\_TransferFunction\_setLocTFs(harkio\_TransferFunction \*tf,**

**harkio\_Matrix \*\*tfs)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_TransferFunction \*tf | 伝達関数 |
| IN | harkio\_Matrix 　\*\*tfs | 定位伝達関数 |
|  |  |  |
|  |  |  |

* **戻り値**

成功 ：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_setSepTFs(harkio\_TransferFunctioに分離用伝達関数を設定する)**

* **機能**

harkio\_TransferFunctionのseptfsに、harkio\_Matrix\*\*（分離用伝達関数） を設定する。

* **構文**

**int harkio\_TransferFunction\_setSepTFs(harkio\_TransferFunction \*tf,**

**harkio\_Matrix \*\*tfs)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_TransferFunction \*tf |  |
| IN | harkio\_Matrix \*\*tfs |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功 ：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_TransferFunction\_write（伝達関数をzipに出力する）**

* **機能**

伝達関数をzipに出力する

* **構文**

**int harkio\_TransferFunction\_write(harkio\_TransferFunction \*tf, char \*path)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf | 伝達関数 |
| OUT | char \*path |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

## **harkio\_TransferFunction\_getTFByPos(許容差範囲内のpositionの伝達関数を取得する)**

* **機能**

許容差範囲内のpositionの伝達関数を取得する

* **構文**

harkio\_Matrix\*

arkio\_Matrix\* harkio\_TransferFunction\_getTFByPos(harkio\_TransferFunction \*tf,

harkio\_Position \*pos,

float tolerance

int type)

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf | 対象harkio\_TransferFunction |
| IN | harkio\_Position \*pos | 対象position |
| IN | float tolerance | 許容差 |
| IN | int type | 伝達関数種別  loctf　：　定位伝達関数,  septf　：　分離伝達関数 |

* **戻り値**

**hark\_Matrix\***

* **処理説明**

tf->poses

0

1

2

3

2

4

6

7

id=2の伝達関数

配列番号

id番号

対になっている

\*\*loctfs/\*\*septfs

## **harkio\_TransferFunction\_getTFById (同じid番号の伝達関数を取得する)**

* **機能**

同じid番号の伝達関数を取得する

* **構文**

**harkio\_Matrix\* harkio\_TransferFunction\_getTFById (harkio\_TransferFunction \*tf,**

**int id ,**

**int type)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
| IN | int id | 取得対象 id |
| IN | int Type | 伝達関数種別  loctf　：　定位伝達関数,  septf　：　分離伝達関数 |
|  |  |  |

* **戻り値**
* **処理説明**

## harkio\_TransferFunction\_getNeighborTFsById (対象IDの近接ID郡の伝達関数を取得する)

* **機能**

対象IDの近接ID郡の伝達関数を取得する

* **構文**

**harkio\_Matrix\*\* harkio\_TransferFunction\_getNeighborTFsById(harkio\_TransferFunction \*tf,**

**int id**

**int type)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_TransferFunction \*tf |  |
| IN | int id | 対象ID |
| IN | int type | 伝達関数種別  loctf　：　定位伝達関数,  septf　：　分離伝達関数 |
|  |  |  |

* **戻り値**

**hark\_Matrix\*\* ：**

* **処理説明**

１

３

７

９

TF

TF

TF

TF

TF

TF

近接ID

伝達関数

入力ID

hark\_Matrix\*\*

出力

## **readZippedXML(zipファイルからharkio\_XMLに変換する)**

* **機能**

Zipファイルを読み込んで、harkio\_XML形式に変換する

* **構文**

**harkio\_XML\* readZippedXML(struct zip \*handle, char \*path)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_TransferFunction \*tf |  |
| IN | Int id |  |
|  |  |  |
|  |  |  |

* **戻り値**
* **処理説明**

## **readZippedText (ZIPファイルを読み込む)**

* **機能**

ZIPファイルを読み込む

* **構文**

**char\* readZippedText(struct zip \*handle, char \*path)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | struct zip \*handle |  |
| IN | char \*path |  |
|  |  |  |
|  |  |  |

* **戻り値**

Char\* : 動的メモリ領域に読み込んだZipファイルの内容

* **処理説明**

## **int libharkio3\_TransferFunction\_readlibhakio2 (libharkio2->libhark3変換)**

* **機能**

libhark2形式の伝達関数からlibhark3形式に変換

* **構文**

**int libharkio3\_TransferFunction\_readlibhakio2(**

**harkio\_TransferFunction \*\*arg\_tf,**

**char \*loctfpath ,**

**char \*septfpath)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_TransferFubction \*\*arg\_tf | 変換後のharkio\_TransferFubction形式 |
| IN | char \*loctfpath | 定位伝達関数ファイルのパス（無い場合はNULL） |
| IN | char \*septfpath | 分離伝達関数ファイルのパス（無い場合はNULL） |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

失敗　：　EXIT\_FAILURE

* **処理説明**

# **Config (全体設定)**

## **harkio\_Config\_new (hark\_Config型領域を新規作成する)**

* **機能**

新規で、hark\_Config型領域を作成する

* **構文**

**harkio\_Config\* harkio\_Config\_new(void)**

* **パラメータ**

なし

* **戻り値**

[**harkio\_Config\***](#_harkio_Config)

* **処理説明**

## **harkio\_Config\_delete (harkio\_Config型領域を削除する)**

* **機能**

harkio\_Config型領域を削除する

* **構文**

**void harkio\_Config\_delete(harkio\_Config \*\*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Config\_getComment (Harkio\_Configからcommentを取得する)**

* **機能**

Harkio\_Configからcommentを取得する

* **構文**

**char\* harkio\_Config\_getComment(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

Char\* ：　comment

* **処理説明**

## **harkio\_Config\_getSynchronousAverage (Harkio\_ConfigからSynchronousAverageを取得する)**

* **機能**

Harkio\_ConfigからSynchronousAverageを取得する

* **構文**

**int harkio\_Config\_getSynchronousAverage(harkio\_Config \*cfg,)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

Int ：　SynchronousAverage

* **使用関数**
* **処理説明**

## **harkio\_Config\_getTSPpath (Harkio\_ConfigからTSPpathを取得する)**

* **機能**

Harkio\_ConfigからTSPpathを取得する

* **構文**

**int harkio\_Config\_getTSPpath(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

char\* cfg->TSPpath

* **処理説明**

## **harkio\_Config\_getTSPoffset (Harkio\_ConfigからTSPoffsetを取得する)**

* **機能**

Harkio\_ConfigからTSPoffsetを取得する

* **構文**

**int harkio\_Config\_getTSPoffset(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

int ：　TSPoffset

* **処理説明**

## **harkio\_Config\_getPeakSearchFrom (Harkio\_ConfigからPeakSearchFromを取得する)**

* **機能**

Harkio\_ConfigからPeakSearchFromを取得する

* **構文**

**int harkio\_Config\_getPeakSearchFrom(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

int ：　PeakSearchFrom(開始サンプル値)

* **処理説明**

## **harkio\_Config\_getPeakSearchTo (Harkio\_ConfigからPeakSearchToを取得する)**

* **機能**

Harkio\_ConfigからPeakSearchToを取得する

* **構文**

**int harkio\_Config\_getPeakSearchTo(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

int ：　PeakSearchFrom(終了サンプル値)

* **処理説明**

## **harkio\_Config\_getNfft (Harkio\_ConfigからNfftを取得する)**

* **機能**

Harkio\_ConfigからNfftを取得する

* **構文**

**int harkio\_Config\_getNfft(harkio\_Config \*cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

int ：　Nfft

* **処理説明**

## **harkio\_Config\_setSynchronousAverage (Harkio\_Config にSynchronousAverageをセットする)**

* **機能**

Harkio\_ConfigにSynchronousAverageをセットする

* **構文**

**int harkio\_Config\_setSynchronousAverage(harkio\_Config \*cfg, int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Config \*\*cfg |  |
| IN | Int value |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setTSPpath (Harkio\_ConfigにTSPpathをセットする)**

* **機能**

Harkio\_ConfigにTSPpathをセットする

* **構文**

**int harkio\_Config\_setTSPoffset(harkio\_Config \*cfg, int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Config \*\*cfg |  |
| IN | Int vlue |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setTSPoffset (Harkio\_ConfigにTSPoffsetをセットする)**

* **機能**

Harkio\_ConfigにTSPoffsetをセットする

* **構文**

**int harkio\_Config\_setTSPpath(harkio\_Config \*cfg, const char \*value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Config \*\*cfg |  |
| IN | Int vlue |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setPeakSearchFrom (hrkio\_ConfigにPeakSearchFromをセットする)**

* **機能**

hrkio\_ConfigにPeakSearchFromをセットする

* **構文**

**int harkio\_Config\_setPeakSearchFrom(harkio\_Config \*cfg, int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | name | Description |
| IN | harkio\_Config \*\*cfg |  |
| OUT | Int value |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setPeakSearchTo (hrkio\_ConfigにPeakSearchToをセットする)**

* **機能**

hrkio\_ConfigにPeakSearchToをセットする

* **構文**

**int harkio\_Config\_setPeakSearchTo(harkio\_Config \*cfg, int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
|  | harkio\_Config \*\*cfg |  |
|  | Int value |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setNfft (Harkio\_ConfigにNfftをセットする)**

* **機能**

Harkio\_ConfigにNfftをセットする

* **構文**

**int harkio\_Config\_setNfft(harkio\_Config \*cfg , int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Config \*\*cfg |  |
| IN | int value |  |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_setComment(Harkio\_ConfigにCommentをセットする)**

* **機能**

Harkio\_ConfigにCommentをセットする

* **構文**

**int harkio\_Config\_setNfft(harkio\_Config \*cfg , int value)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| OUT | harkio\_Config \*\*cfg |  |
| IN | int value | セットするComment値 |
|  |  |  |
|  |  |  |

* **戻り値**

成功　：　EXIT\_SUCCESS

* **処理説明**

## **harkio\_Config\_isEqual（ harkio\_Configの比較）**

* **機能**

入力されたharkio\_Configの比較を行う

* **構文**

**int harkio\_Config\_isEqual(harkio\_Config \*a, harkio\_Config \*b)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Position \*a |  |
| IN | harkio\_Position \*b |  |
|  |  |  |
|  |  |  |

* **戻り値**

０　：　不一致

1　：　一致

* **処理説明**

## **harkio\_Config\_prin　（harkio\_Configの内容を出力する）**

* **機能**

harkio\_Configの内容を出力する

* **構文**

**void harkio\_Config\_print(harkio\_Config\* cfg)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg |  |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Config\_copy (harkio\_Configの内容をコピーする)**

* **機能**

harkio\_Configの内容をコピーする

* **構文**

**harkio\_Config\* harkio\_Config\_copy(harkio\_Config \*cfg1)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | harkio\_Config \*\*cfg1 | コピー元のharkio\_config |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

[**harkio\_Config\***](#_harkio_Config) **：**コピー先のharkio\_config

* **処理説明**

# **LOG (ログ出力)**

## **harkio\_Log\_setLevel (printlevelを設定する)**

* **機能**

ログ表示レベルを設定する

ログ表示レベルは、下記の3種類ある。

debug

warnning

error

初期値は、errorレベル

* **構文**

**void harkio\_Log\_setLevel(int PrintLevel)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | int PrintLevel | harkio\_Log\_Error : エラー表示  harkio\_Log\_Warning : 警告表示  harkio\_Log\_Debug : デバッグ表示 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Log\_printf (ログを表示する)**

* **機能**

ログを表示する

* **構文**

**void harkio\_Log\_printf(enum harkio\_Log\_logLevel level,**

**char \*message)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | enum harkio\_Log\_logLevel LogLevel | harkio\_Log\_Error : エラー  harkio\_Log\_Warning : 警告  harkio\_Log\_Debug : デバッグ |
|  | char \*message |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **void harkio\_Log\_FileOpenFailed (ファイルオープンエラーログを表示する)**

* **機能**

ファイルオープンエラーのログを表示する

* **構文**

**void harkio\_Log\_FileOpenFailed(char \*filename)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char\* filename | ログに出力するファイル名 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Log\_printf (valueが不正の場合のエラーログを表示する)**

* **機能**

valueが不正の場合のエラーログを表示する

* **構文**

**void harkio\_Log\_WrongValue(char \*valuename, char \*reason)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*valuename, | valueの名前 |
| IN | char \*reason | エラーの詳細 |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**

## **harkio\_Log\_NullPointer (pointerがNULLの場合のエラーログを表示する)**

* **機能**

PointerがNULLの場合のエラーログを表示する

* **構文**

**void harkio\_Log\_NullPointer(char \*valuename)**

* **パラメータ**

|  |  |  |
| --- | --- | --- |
| I/O | Name | Description |
| IN | char \*valuename, | valueの名前 |
|  |  |  |
|  |  |  |
|  |  |  |

* **戻り値**

なし

* **処理説明**